

Pearson

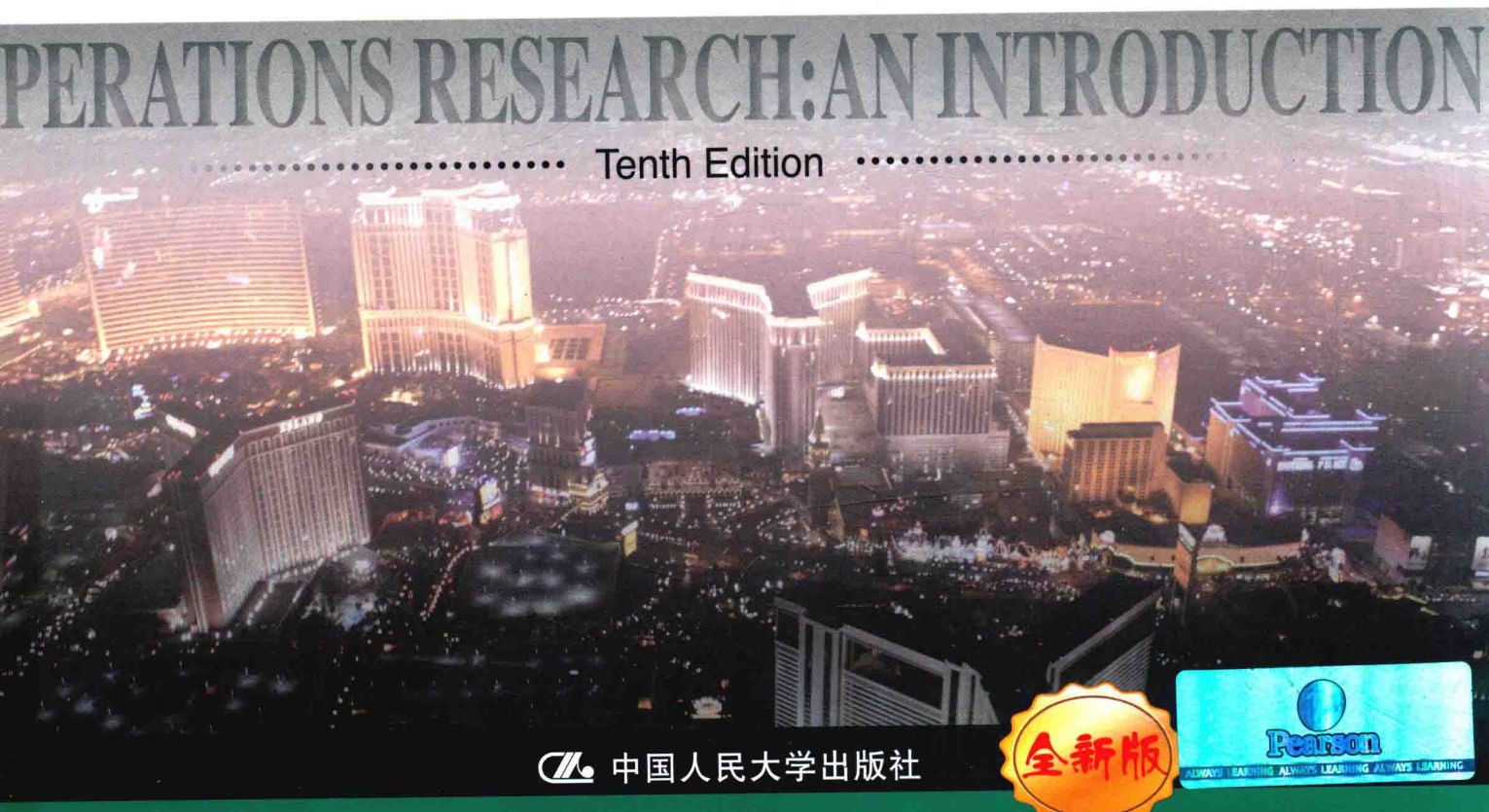
管理科学与工程经典丛书

MANAGEMENT SCIENCE AND ENGINEERING CLASSICS

运筹学基础

英文版·第10版

哈姆迪·塔哈（Hamdy A. Taha） 著



需要全本请访问 www.ertongbook.com

管理科学与工程经典丛书

MANAGEMENT SCIENCE AND ENGINEERING CLASSICS

运筹学基础

英文版·第10版

哈姆迪·塔哈（Hamdy A. Taha） 著

PERATIONS RESEARCH: AN INTRODUCTION

Tenth Edition

中国人民大学出版社

·北京·

图书在版编目（CIP）数据

运筹学基础：第10版：英文/哈姆迪·塔哈著. —北京：中国人民大学出版社，2019.1
管理科学与工程经典丛书
ISBN 978-7-300-26637-4

Ⅰ.①运… Ⅱ.①哈… Ⅲ.①运筹学-高等学校-教材-英文 Ⅳ.①O22

中国版本图书馆CIP数据核字（2019）第003982号

管理科学与工程经典丛书
运筹学基础（英文版·第10版）
哈姆迪·塔哈 著

出版发行	中国人民大学出版社		
社　　址	北京中关村大街31号	**邮政编码**	100080
电　　话	010-62511242（总编室）	010-62511770（质管部）	
	010-82501766（邮购部）	010-62514148（门市部）	
	010-62515195（发行公司）	010-62515275（盗版举报）	
网　　址	http:// www. crup. com. cn		
	http:// www. ttrnet. com（人大教研网）		
经　　销	新华书店		
印　　刷	三河市恒彩印务有限公司		
规　　格	215 mm × 275 mm 16开本	**版　　次**	2019年1月第1版
印　　张	31.5 插页2	**印　　次**	2019年1月第1次印刷
字　　数	920 000	**定　　价**	65.00元

版权所有　侵权必究　印装差错　负责调换

总　序

随着我国加入WTO，越来越多的国内企业参与到国际竞争中来，用国际上通用的语言思考、工作、交流的能力也越来越受到重视。这样一种能力也成为我国各类人才参与竞争的一种有效工具。国家教育机构、各类院校以及一些主要的教材出版单位一直在思考，如何顺应这一发展潮流，推动各层次人员通过学习来获取这种能力。双语教学就是这种背景下的一种尝试。

双语教学在我国主要指汉语和国际通用的英语教学。事实上，双语教学在我国教育界已经不是一个陌生的词汇了，以双语教学为主的科研课题也已列入国家“十五”规划的重点课题。但从另一方面来看，双语教学从其诞生的那天起就被包围在人们的赞成与反对声中。如今，依然是有人赞成有人反对，但不论是赞成居多还是反对占上，双语教学的规模和影响都在原有的基础上不断扩大，且呈大发展之势。一些率先进行双语教学的院校在实践中积累了经验，不断加以改进；一些待进入者也在模仿中学习，并静待时机成熟时加入这一行列。由于我国长期缺乏讲第二语言（包括英语）的环境，开展双语教学面临特殊的困难，因此，选用合适的教材就成为双语教学成功与否的一个重要问题。我们认为，双语教学从一开始就应该使用原版的各类学科的教材，而不是由本土教师自编的教材，从而可以避免中国式英语问题，保证语言的原汁原味。各院校除应执行国家颁布的教学大纲和课程标准外，还应根据双语教学的特点和需要，适当调整教学课时的设置，合理选择优秀的、合适的双语教材。

顺应这样一种大的教育发展趋势，中国人民大学出版社同众多国际知名的大出版公司，如麦格劳-希尔出版公司、培生教育出版公司等合作，面向大学本科生层次，遴选了一批国外最优秀的管理类原版教材，涉及专业基础课，人力资源管理、市场营销及国际化管理等专业方向课，并广泛听取有着丰富的双语一线教学经验的教师的建议和意见，对原版教材进行了适当的改编，删减了一些不适合我国国情和不适合教学的内容；另一方面，根据教育部对双语教学教材篇幅合理、定价低的要求，我们更是努力区别于目前市场上形形色色的各类英文版、英文影印版的大部头，将目标受众锁定在大学本科生层次。本套教材尤其突出了以下一些特点：

● 保持英文原版教材的特色。本套双语教材根据国内教学实际需要，对原书进行了一定的改编，主要是删减了一些不适合教学以及不符合我国国情的内容，但在体系结构和内容特色方面都保持了原版教材的风貌。专家们的认真改编和审定，使本套教材既保持了学术上的完整性，又贴近中国实际；既方便教师教学，又方便学生理解和掌握。

● 突出管理类专业教材的实用性。本套教材既强调学术的基础性，又兼顾应用的广泛性；既侧重让学生掌握基本的理论知识、专业术语和专业表达方式，又考虑到教材和管理实践的紧密结合，有助于学生形成专业的思维能力，培养实际的管理技能。

● 体系经过精心组织。本套教材在体系架构上充分考虑到当前我国在本科教育阶段推广双语教学的进度安排，首先针对那些课程内容国际化程度较高的学科进行双语教材开发，在其专业模块内精心选择各专业教材。这种安排既有利于我国教师摸索双语教学的经验，使得双语教学贴近现实教学的需要；也有利于我们收集关于双语教学教材的建议，更好地推出后续的双语教材及教辅材料。

● 篇幅合理，价格相对较低。为适应国内双语教学内容和课时上的实际需要，本套教材进行了一定的删减和改编，使总体篇幅更为合理；而采取低定价，则充分考虑到了学生实际的购买能力，从而使本套教材得以真正走近广大读者。

● 提供强大的教学支持。依托国际大出版公司的力量，本套教材为教师提供了配套的教辅材料，如教师手册、PowerPoint 讲义、试题库等，并配有内容极为丰富的网络资源，从而使教学更为便利。

本套教材是在双语教学教材出版方面的一种尝试。我们在选书、改编及出版的过程中得到了国内许多高校的专家、教师的支持和指导，在此深表谢意。同时，为使我们后续推出的教材更适于教学，我们也真诚地期待广大读者提出宝贵的意见和建议。需要说明的是，尽管我们在改编的过程中已加以注意，但由于各教材的作者所处的政治、经济和文化背景不同，书中内容仍可能有不妥之处，望读者在阅读时注意比较和甄别。

徐二明

中国人民大学商学院

需要全本请访问 www.ertongbook.com

What's New in the Tenth Edition*

Over the past few editions, I agonized over the benefit of continuing to include the hand computational algorithms that, to my thinking, have been made obsolete by present-day great advances in computing. I no longer have this "anxiety" because I sought and received feedback from colleagues regarding this matter. The consensus is that these classical algorithms must be preserved because they are an important part of OR history. Some responses even included possible scenarios (now included in this edition) in which these classical algorithms can be beneficial in practice.

In the spirit of my colleagues collective wisdom, which I now enthusiastically espouse, I added throughout the book some 25 entries titled *Aha! moments.* These entries, written mostly in an informal style, deal with OR anecdotes/stories (some dating back to centuries ago) and OR concepts (theory, applications, computations, and teaching methodology). The goal is to provide a historical perspective of the roots of OR (and, hopefully, render a "less dry" book read).

Additional changes/additions in the tenth edition include:

- Using a brief introduction, inventory modeling is presented within the more encompassing context of supply chains.
- New sections are added about computational issues in the simplex method and in inventory.
- This edition adds two new case analyses, resulting in a total of 17 fully developed real-life applications. All the cases appear in Chapter 26 on the website.
- By popular demand, all problems now appear at end of their respective chapters and are cross-referenced by text section to facilitate making problem assignments.
- New problems have been added.
- TORA software has been updated.

---

* 本书篇幅过大，章节有删减，章节顺序作了相应调整，但网上（www.pearsonhighered.com/taha）文件保持原状，文件序号未作更改。

Acknowledgments

I salute my readership—students, professors, and OR practitioners—for their confidence and support over the past 45 years. I am especially grateful to the following colleagues who provided helpful comments in response to my proposal for the tenth edition: Bhaba Sarker (Louisiana State University), Michael Fraboni (Moravian College), Layek Abdel-Malek (New Jersey Institute of Technology), James Smith (University of Massachusetts), Hansuk Sohn (New Mexico State University), Elif Kongar (University of Bridgeport), Sung, Chung-Hsien (University of Illinois), Kash A. Barker (University of Oklahoma).

Professor Michael Trick (Carnegie Mellon University) provided insightful arguments regarding the importance of continuing to include the classical (hand-computational) algorithms of yore in the book and I now enthusiastically share the essence of his statement that "[He] would not be happy to see the day when the Hungarian algorithm is lost to our textbooks."

I wish to thank Professor Donald Erlenkotter (University of California, Los Angles) for his feedback on material in the inventory chapters and Professor Xinhui Zhang (Wright State University) for his input during the preparation of the inventory case study. I also wish to thank Professors Hernan Abeledo (The George Washington University), Ali Diabat (Masdar Institute of Science and Technology, Abu Dhabi, UAE), Robert E. Lewis (University of Alabama, Huntsville), Scott Long (Liberty University), and Daryl Santos (Binghamton University) for pointing out discrepancies in the ninth edition and making suggestions for the tenth.

I offer special thanks and appreciation to the editorial and production teams at Pearson for their valuable help during the preparation of this edition: Marcia Horton (Vice President/Editorial Director, Engineering, Computer Science), Holly Stark (Executive Editor), Scott Disanno (Senior Manager Editor), George Jacob (Project Manager), Erin Ault (Program Manager), Amanda Brands (Editorial Assistant).

It is a great pleasure to recognize Jack Neifert, the first acquisition editor with my former publisher Macmillan, who in 1972, one year after the publication of the first edition, predicted that "this is a book with a long life." The tenth edition is an apt testimonial to the accuracy of Jack's prediction.

I am grateful to Tamara Ellenbecker, Carrie Pennington, Matthew Sparks and Karen Standly, all of the University of Arkansas Industrial Engineering Department, for their able help (and patience) during the preparation of this edition.

My son Sharif, though a neuroscientist, has provided an insightful critique of the *Aha! moments* in this edition.

HAMDY A. TAHA
hat@uark.edu

简明目录

第1章　什么是运筹学 1

第2章　线性规划建模 12

第3章　单纯形法和灵敏度分析 50

第4章　对偶性与后最优分析 99

第5章　各种运输模型 127

第6章　网络模型 157

第7章　目标规划 200

第8章　整数线性规划 213

第9章　启发式规划 237

第10章　确定性动态规划 266

第11章　库存模型（供应链介绍） 291

第12章　决策分析与博弈 323

第13章　随机库存模型 355

第14章　排队系统 369

第15章　仿真模型 410

第16章　经典最优化理论 433

第17章　非线性规划算法 450

附录A　部分习题答案 474

附录B　统计表 487

Contents

Chapter 1 What Is Operations Research? 1

1.1 Introduction *1*
1.2 Operations Research Models *1*
1.3 Solving the OR Model *3*
1.4 Queuing and Simulation Models *5*
1.5 Art of Modeling *5*
1.6 More than Just Mathematics *6*
1.7 Phases of an OR Study *8*
1.8 About this Book *9*
Bibliography *10*
Problems *10*

Chapter 2 Modeling with Linear Programming 12

2.1 Two-Variable LP Model *12*
2.2 Graphical LP Solution *14*
2.3 Computer Solution with Solver and AMPL *18*
2.4 Linear Programming Applications *24*
Bibliography *39*
Problems *39*

Chapter 3 The Simplex Method and Sensitivity Analysis 50

3.1 LP Model in Equation Form *50*
3.2 Transition from Graphical to Algebraic Solution *51*
3.3 The Simplex Method *54*
3.4 Artificial Starting Solution *61*
3.5 Special Cases in the Simplex Method *66*
3.6 Sensitivity Analysis *71*
3.7 Computational Issues in Linear Programming *83*
Bibliography *86*
Case Study: Optimization of Heart Valves Production *86*
Problems *89*

Chapter 4 Duality and Post-Optimal Analysis 99

4.1 Definition of the Dual Problem *99*
4.2 Primal–Dual Relationships *102*

4.3 Economic Interpretation of Duality *107*
4.4 Additional Simplex Algorithms *110*
4.5 Post-Optimal Analysis *113*
Bibliography *119*
Problems *119*

Chapter 5 Transportation Model and Its Variants 127
5.1 Definition of the Transportation Model *127*
5.2 Nontraditional Transportation Models *131*
5.3 The Transportation Algorithm *133*
5.4 The Assignment Model *144*
Bibliography *148*
Case Study: Scheduling Appointments at Australian Tourist Commission Trade Events *148*
Problems *152*

Chapter 6 Network Model 157
6.1 Scope and Definition of Network Models *157*
6.2 Minimal Spanning Tree Algorithm *159*
6.3 Shortest-Route Problem *161*
6.4 Maximal Flow Model *173*
6.5 CPM and Pert *180*
Bibliography *190*
Case Study: Saving Federal Travel Dollars *190*
Problems *193*

Chapter 7 Goal Programming 200
7.1 A Goal Programming Formulation *200*
7.2 Goal Programming Algorithms *202*
Bibliography *208*
Case Study: Allocation of Operating Room Time in Mount Sinai Hospital *208*
Problems *211*

Chapter 8 Integer Linear Programming 213
8.1 Illustrative Applications *213*
8.2 Integer Programming Algorithms *219*
Bibliography *229*
Problems *230*

Chapter 9 Heuristic Programming 237
9.1 Introduction *237*
9.2 Greedy (Local Search) Heuristics *238*
9.3 Metaheuristic *243*
9.4 Application of Metaheuristics to Integer Linear Programs *253*
9.5 Introduction to Constraint Programming (CP) *259*

Bibliography *261*
Problems *261*

Chapter 10 Deterministic Dynamic Programming 266

10.1 Recursive Nature of Dynamic Programming (DP) Computations *266*
10.2 Forward and Backward Recursion *269*
10.3 Selected DP Applications *271*
10.4 Problem of Dimensionality *283*
Bibliography *285*
Case Study: Optimization of Crosscutting and Log Allocation at Weyerhaeuser *285*
Problems *288*

Chapter 11 Inventory Modeling (with Introduction to Supply Chains) 291

11.1 Inventory Problem: A Supply Chain Perspective *291*
11.2 Role of Demand in the Development of Inventory Models *294*
11.3 Static Economic-Order-Quantity Models *296*
11.4 Dynamic EOQ Models *304*
11.5 Sticky Issues in Inventory Modeling *316*
Bibliography *317*
Case Study: Kroger Improves Pharmacy Inventory Management *317*
Problems *320*

Chapter 12 Decision Analysis and Games 323

12.1 Decision Making Under Certainty—Analytic Hierarchy Process (AHP) *323*
12.2 Decision Making Under Risk *329*
12.3 Decision Under Uncertainty *335*
12.4 Game Theory *338*
Bibliography *345*
Case Study: Booking Limits in Hotel Reservations *345*
Problems *347*

Chapter 13 Probabilistic Inventory Models 355

13.1 Continuous Review Models *355*
13.2 Single-Period Models *360*
13.3 Multiperiod Model *365*
Bibliography *366*
Problems *367*

Chapter 14 Queuing Systems 369

14.1 Why Study Queues? *369*
14.2 Elements of a Queuing Model *370*
14.3 Role of Exponential Distribution *372*
14.4 Pure Birth and Death Models (Relationship Between the Exponential and Poisson Distributions) *373*

需要全本请访问 www.ertongbook.com

14.5 General Poisson Queuing Model *377*
14.6 Specialized Poisson Queues *379*
14.7 (*M*/*G*/1):(*GD*/∞/∞)—Pollaczek-Khintchine (P-K) Formula *394*
14.8 Other Queuing Models *395*
14.9 Queuing Decision Models *396*
Bibliography *400*
Case Study: Analysis of an Internal Transport System in a Manufacturing Plant *400*
Problems *402*

Chapter 15 Simulation Modeling 410
15.1 Monte Carlo Simulation *410*
15.2 Types of Simulation *413*
15.3 Elements of Discrete Event Simulation *413*
15.4 Generation of Random Numbers *418*
15.5 Mechanics of Discrete Simulation *419*
15.6 Methods for Gathering Statistical Observations *424*
15.7 Simulation Languages *427*
Bibliography *429*
Problems *429*

Chapter 16 Classical Optimization Theory 433
16.1 Unconstrained Problems *433*
16.2 Constrained Problems *438*
Bibliography *448*
Problems *448*

Chapter 17 Nonlinear Programming Algorithms 450
17.1 Unconstrained Algorithms *450*
17.2 Constrained Algorithms *455*
Bibliography *472*
Problems *472*

Appendix A Partial Answers to Selected Problems 474

Appendix B Statistical Tables 487

CHAPTER 1

What Is Operations Research?

1.1 INTRODUCTION

The first formal activities of Operations Research (OR) were initiated in England during World War II, when a team of British scientists set out to assess the best utilization of war materiel based on scientific principles rather than on ad hoc rules. After the war, the ideas advanced in military operations were adapted to improve efficiency and productivity in the civilian sector.

This chapter introduces the basic terminology of OR, including mathematical modeling, feasible solutions, optimization, and iterative algorithmic computations. It stresses that defining the problem correctly is the most important (and most difficult) phase of practicing OR. The chapter also emphasizes that, while mathematical modeling is a cornerstone of OR, unquantifiable factors (such as human behavior) must be accounted for in the final decision. The book presents a variety of applications using solved examples and chapter problems. In particular, the book includes end-of-chapter fully developed case analyses.

1.2 OPERATIONS RESEARCH MODELS

Consider the following **tickets purchasing problem**. A businessperson has a 5-week commitment traveling between Fayetteville (FYV) and Denver (DEN). Weekly departure from Fayetteville occurs on Mondays for return on Wednesdays. A regular roundtrip ticket costs $400, but a 20% discount is granted if the roundtrip dates span a weekend. A one-way ticket in either direction costs 75% of the regular price. How should the tickets be bought for the 5-week period?

We can look at the situation as a decision-making problem whose solution requires answering three questions:

1. What are the decision **alternatives**?
2. Under what **restrictions** is the decision made?
3. What is an appropriate **objective criterion** for evaluating the alternatives?

Three plausible alternatives come to mind:

1. Buy five regular FYV-DEN-FYV for departure on Monday and return on Wednesday of the same week.

2. Buy one FYV-DEN, four DEN-FYV-DEN that span weekends, and one DEN-FYV.
3. Buy one FYV-DEN-FYV to cover Monday of the first week and Wednesday of the last week and four DEN-FYV-DEN to cover the remaining legs. All tickets in this alternative span at least one weekend.

The restriction on these options is that the businessperson should be able to leave FYV on Monday and return on Wednesday of the same week.

An obvious objective criterion for evaluating the proposed alternatives is the price of the tickets. The alternative that yields the smallest cost is the best. Specifically, we have:

$$\text{Alternative 1 cost} = 5 \times \$400 = \$2000$$

$$\text{Alternative 2 cost} = .75 \times \$400 + 4 \times (.8 \times \$400) + .75 \times \$400 = \$1880$$

$$\text{Alternative 3 cost} = 5 \times (.8 \times \$400) = \mathbf{\$1600}$$

Alternative 3 is the cheapest.

Though the preceding example illustrates the three main components of an OR model—alternatives, objective criterion, and constraints—situations differ in the details of how each component is developed, and how the resulting model is solved. To illustrate this point, consider the following **garden problem**: A home owner is in the process of starting a backyard vegetable garden. The garden must take on a rectangular shape to facilitate row irrigation. To keep critters out, the garden must be fenced. The owner has enough material to build a fence of length $L = 100$ ft. The goal is to fence the largest possible rectangular area.

In contrast with the tickets example, where the number of alternatives is finite, the number of alternatives in the present example is infinite; that is, the *width* and *height* of the rectangle can each assume (theoretically) infinity of values between 0 and L. In this case, the width and the height are **continuous variables**.

Because the variables of the problem are continuous, it is impossible to find the solution by exhaustive enumeration. However, we can *sense* the trend toward the best value of the garden area by fielding increasing values of width (and hence decreasing values of height). For example, for $L = 100$ ft, the combinations (width, height) = (10, 40), (20, 30), (25, 25), (30, 20), and (40, 10) respectively yield (area) = (400, 600, 625, 600, and 400), which demonstrates, but not proves, that the largest area occurs when width = height = $L/4 = 25$ ft. Clearly, this is no way to compute the optimum, particularly for situations with several decision variables. For this reason, it is important to express the problem mathematically in terms of its unknowns, in which case the best solution is found by applying appropriate solution methods.

To demonstrate how the *garden problem* is expressed mathematically in terms of its two unknowns, width and height, define

$$w = \text{width of the rectangle in feet}$$

$$h = \text{height of the rectangle in feet}$$

Based on these definitions, the restrictions of the situation can be expressed verbally as

1. Width of rectangle + Height of rectangle = Half the length of the garden fence
2. Width and height cannot be negative

These restrictions are translated algebraically as

1. $2(w + h) = L$
2. $w \geq 0, h \geq 0$

The only remaining component now is the objective of the problem; namely, maximization of the area of the rectangle. Let z be the area of the rectangle, then the complete model becomes

$$\text{Maximize } z = wh$$

subject to

$$2(w + h) = L$$

$$w, h \geq 0$$

Actually, this model can be simplified further by eliminating one of the variables in the objective function using the constraint equation; that is,

$$w = \tfrac{L}{2} - h$$

The result is

$$z = wh = (\tfrac{L}{2} - h)h = \tfrac{Lh}{2} - h^2$$

The maximization of z is achieved by using differential calculus (Chapter 16), which yields the best solution as $h = \frac{L}{4} = 25$ ft. Back substitution in the constraint equation then yields $w = \frac{L}{4} = 25$ ft. Thus the solution calls for constructing a square-shaped garden.

Based on the preceding two examples, the general OR model can be organized in the following general format:

Maximize or minimize **Objective Function**

subject to

Constraints

A solution is **feasible** if it satisfies all the constraints. It is **optimal** if, in addition to being feasible, it yields the best (maximum or minimum) value of the objective function. In the *ticket purchasing problem*, the problem considers three feasible alternatives, with the third alternative being optimal. In the *garden problem*, a feasible alternative must satisfy the condition $w + h = \frac{L}{2}$, with w and $h \geq 0$, that is, **nonnegative variables**. This definition leads to an infinite number of feasible solutions and, unlike the *ticket purchasing problem*, which uses simple price comparisons, the optimum solution is determined using differential calculus.

Though OR models are designed to *optimize* a specific objective criterion subject to a set of constraints, the quality of the resulting solution depends on the degree of completeness of the model in representing the real system. Take, for example, the *ticket purchasing model*. If *all* the dominant alternatives for purchasing the tickets are not identified, then the resulting solution is optimum only relative to the alternatives represented in the model. To be specific, if for some reason alternative 3 is left out of the model, the resulting "optimum" solution would call for purchasing the tickets for $1880, which is a **suboptimal** solution. The conclusion is that "the" optimum solution of a model is best only for *that* model. If the model happens to represent the real system reasonably well, then its solution is optimum also for the real situation.

1.3 SOLVING THE OR MODEL

In practice, OR does not offer a single general technique for solving all mathematical models. Instead, the type and complexity of the mathematical model dictate the nature of the solution method. For example, in Section 1.2 the solution of the *tickets purchas-*

ing problem requires simple ranking of alternatives based on the total purchasing price, whereas the solution of the *garden problem* utilizes differential calculus to determine the maximum area.

The most prominent OR technique is **linear programming**. It is designed for models with linear objective and constraint functions. Other techniques include **integer programming** (in which the variables assume integer values), **dynamic programming** (in which the original model can be decomposed into smaller more manageable subproblems), **network programming** (in which the problem can be modeled as a network), and **nonlinear programming** (in which functions of the model are nonlinear). These are only a few among many available OR tools.

A peculiarity of most OR techniques is that solutions are not generally obtained in (formula-like) closed forms. Instead, they are determined by **algorithms**. An algorithm provides fixed computational rules that are applied repetitively to the problem, with each repetition (called **iteration**) attempting to move the solution closer to the optimum. Because the computations in each iteration are typically tedious and voluminous, it is imperative in practice to use the computer to carry out these algorithms.

Some mathematical models may be so complex that it becomes impossible to solve them by any of the available optimization algorithms. In such cases, it may be necessary to abandon the search for the *optimal* solution and simply seek a *good* solution using **heuristics** or **metaheuristics**, a collection of intelligent search *rules of thumb* that move the solution point advantageously toward the optimum.

Aha! Moment: Ada Lovelace, the First-Ever Algorithm Programmer

Though the first conceptual development of an algorithm is attributed to the founder of algebra Muhammad Ibn-Musa Al-Khwarizmi (born c. 780 in Khuwarezm, Uzbekistan, died c. 850 in Baghdad, Iraq),[1] it was British Ada Lovelace (1815–1852) who developed the first computer algorithm. And when we speak of computers, we are referring to the mechanical Difference and Analytical Engines pioneered and designed by the famed British mathematician Charles Babbage (1791–1871).

Lovelace had a keen interest in mathematics. As a teenager, she visited the Babbage home and was fascinated by his invention and its potential uses in doing more than just arithmetic operations. Collaborating with Babbage, she translated into English an article that provided the design details of the Analytical Engine. The article was based on lectures Babbage presented in Italy. In the translated article, Lovelace appended her own notes (which turned out to be longer than the original article and included some corrections of Babbage's design ideas). One of her notes detailed the first-ever *algorithm*, that of computing Bernoulli numbers on the yet-to-be-completed Analytical Engine. She even predicted that the Babbage machine had the potential to manipulate symbols (and not just numbers) and to create complex music scores.[2]

Ada Lovelace died at the young age of 37. In her honor, the computer language *Ada*, developed for the United States Department of Defense, was named after her. The annual mid-October *Ada Lovelace Day* is an international celebration of women in science, technology, engineering, mathematics (STEM). And those of us who have visited St. James Square in London may recall the blue plaque that read "Ada Countess of Lovelace (1815–1852) Pioneer of Computing."

[1]According to Dictionary.com, the word algorithm originates "from Medieval Latin *algorismus*, a mangled transliteration of Arabic al-Khwarizmi."

[2]Lack of funding, among other factors, prevented Babbage from building fully working machines during his lifetime. It was only in 1991 that the London Science Museum built a complete Difference Engine No. 2 using the same materials and technology available to Babbage, thus vindicating his design ideas. There is currently an ongoing long-term effort to construct a fully working Analytical Engine funded entirely by public contributions. It is impressive that modern-day computers are based on the same principal components (memory, CPU, input, and output) advanced by Babbage 100 years earlier.

1.4 QUEUING AND SIMULATION MODELS

Queuing and simulation deal with the study of waiting lines. They are not optimization techniques; rather, they determine measures of performance of waiting lines, such as average waiting time in queue, average waiting time for service, and utilization of service facilities, among others.

Queuing models utilize probability and stochastic models to analyze waiting lines, and simulation estimates the measures of performance by "imitating" the behavior of the real system. In a way, simulation may be regarded as the next best thing to observing a real system. The main difference between queuing and simulation is that queuing models are purely mathematical, and hence are subject to specific assumptions that limit their scope of application. Simulation, on the other hand, is flexible and can be used to analyze practically any queuing situation.

The use of simulation is not without drawbacks. The process of developing simulation models is costly in both time and resources. Moreover, the execution of simulation models, even on the fastest computer, is usually slow.

1.5 ART OF MODELING

The illustrative models developed in Section 1.2 are exact representations of real situations. This is a rare occurrence in OR, as the majority of applications usually involve (varying degrees of) approximations. Figure 1.1 depicts the levels of abstraction that characterize the development of an OR model. We abstract the assumed real world from the real situation by concentrating on the dominant variables that control the behavior of the real system. The model expresses in an amenable manner the mathematical functions that represent the behavior of the assumed real world.

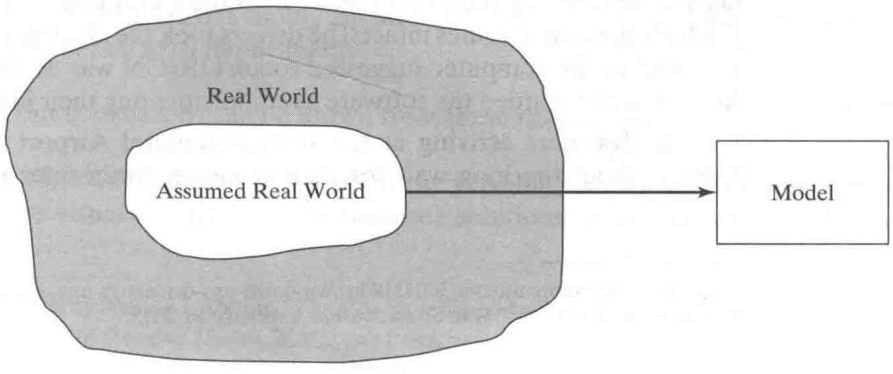
To illustrate levels of abstraction in modeling, consider the Tyko Manufacturing Company, where a variety of plastic containers are produced. When a production order is issued to the production department, necessary raw materials are acquired from the company's stocks or purchased from outside sources. Once a production batch is completed, the sales department takes charge of distributing the product to retailers.

A viable question in the analysis of Tyko's situation is the determination of the size of a production batch. How can this situation be represented by a model?

Looking at the overall system, a number of variables can bear directly on the level of production, including the following (partial) list categorized by department:

1. *Production Department*: Production capacity expressed in terms of available machine and labor hours, in-process inventory, and quality control standards.

FIGURE 1.1
Levels of abstraction in model development



需要全本请访问 www.ertongbook.com